ICLR 2021 Oral

WHEN DO CURRICULA WORK?

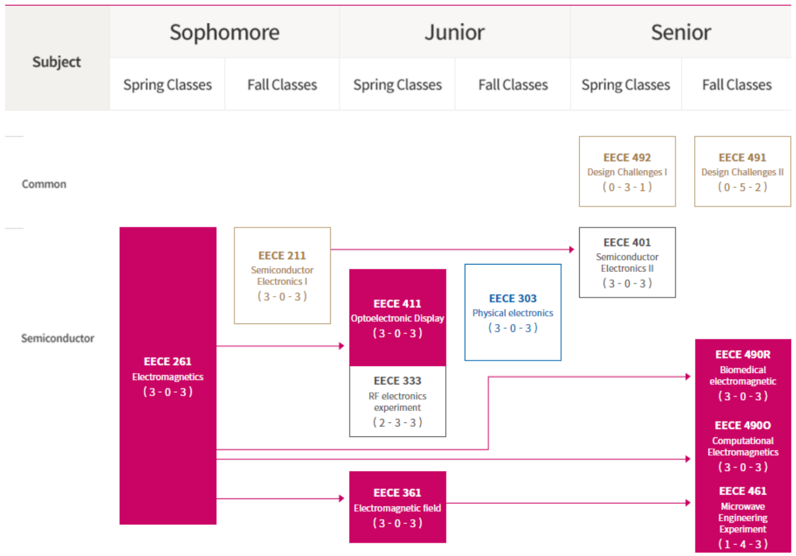
Xiaoxia Wu, Ethan Dyer, Behnam Neyshabur

Efficient Learning Lab@POSTECH
Junwon Seo

Curriculum Learning

- Inspired by the importance of the **ordered** method when teaching humans

- Propose training models by presenting **easier** examples **earlier** during training



Generality of Curriculum Learning

- Anti-curriculum learning
  - Can be as good as curriculum learning in certain scenarios.
  - Underperform standard or curriculum learning in other contexts.

→ Empirical observations on curricula appear to be in conflict

- Despite a rich literature(about 50), do we use it?
  - No ordered learning method is known to improve consistently across contexts.

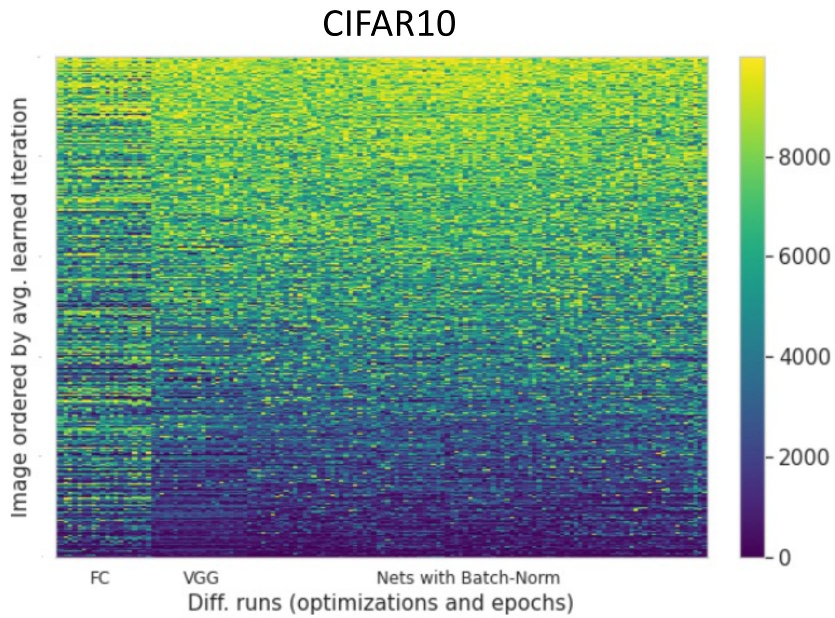
Implicit Curricula

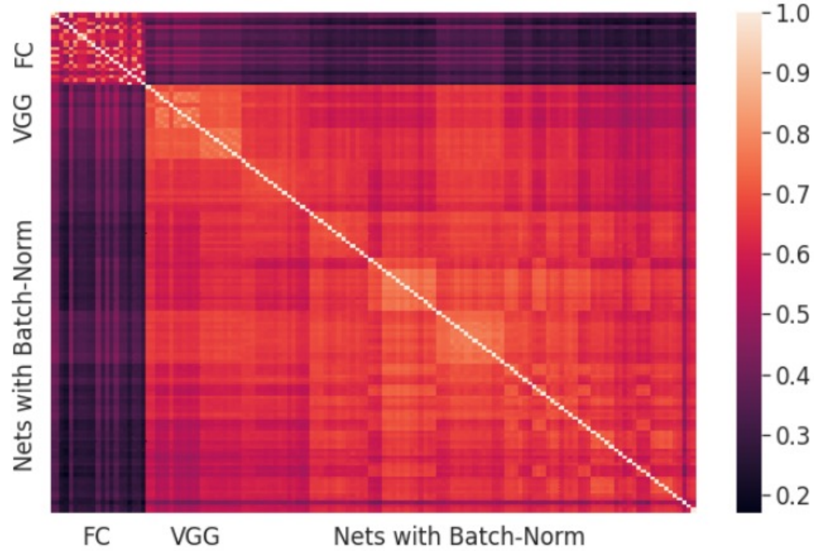
- The order in which a network learns examples under traditional stochastic gradient descent with i.i.d. data sampling

- How to quantify the order?
  - Define the *learned iteration* of a sample for a given model
  - The epoch for which the model correctly predicts the sample and all subsequent epochs
  - Explicitly,

$$\min_{t^*}\{t^*|\hat{y}_{\mathbf{w}}(t)_i = y_i, \forall t^* \leq t \leq T\}$$

Implicit Curricula





- FC/VGG/ResNet/WideResNet/DenseNet/EfficientNet B0/VGG-BN and Adam/SGD with momentum
- “At least within model types, **less ambiguity** about the difficulty of **a given image**”

Three Ingredients in Curriculum learning

- The scoring function
  - A map from an input example, x, to a numerical score $s(\boldsymbol{x}) \in \mathbb{R}$
  - Higher score corresponds to a more difficult example

- The pacing function
  - The size of the training dataset used at each step t.

- The order
  - Curriculum: from **lowest** score to **highest** score
  - Anti-curriculum: from **highest** score to **lowest** score
  - Random

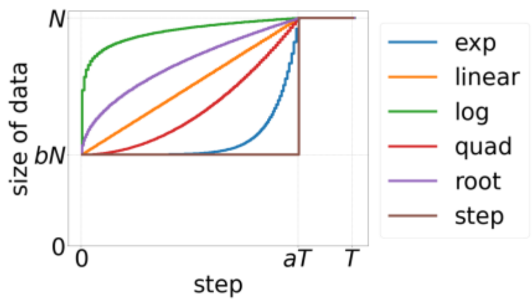
Three Ingredients in Curriculum learning

Algorithm 1 (Random-/Anti-) Curriculum learning with pacing and scoring functions

1: **Input:** Initial weights $\mathbf{w}^0$, training set $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, pacing function $g : [T] \to [N]$, scoring function $s : [N] \to \mathbb{R}$, order $o \in \{$"ascending", "descending", "random"$\}$.
2: $(\mathbf{x}_1, \ldots, \mathbf{x}_N) \leftarrow \text{sort}(\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}, s, o)$
3: **for** $t = 1, \ldots, T$ **do**
4: $\quad \mathbf{w}^{(t)} \leftarrow \text{train-one-epoch}(\mathbf{w}^{(t-1)}, \{\mathbf{x}_1, \ldots, \mathbf{x}_{g(t)}\})$
5: **end for**

Random ordering corresponds to **i.i.d. training** on a training dataset with **dynamic size**

Name	Expression $g_{(a,b)}(t)$
log	$Nb + N(1-b)\left(1 + .1\log\left(\frac{t}{aT} + e^{-10}\right)\right)$
exp	$Nb + \frac{N(1-b)}{e^{10}-1}\left(\exp\left(\frac{10t}{aT}\right) - 1\right)$
step	$Nb + N\lceil \frac{x}{aT} \rceil$
polynomial	$Nb + N\frac{(1-b)}{(aT)^p}t^p \quad - \ p = 1/2, 1, 2$



a: the fraction of training needed to reach the size of the full training set

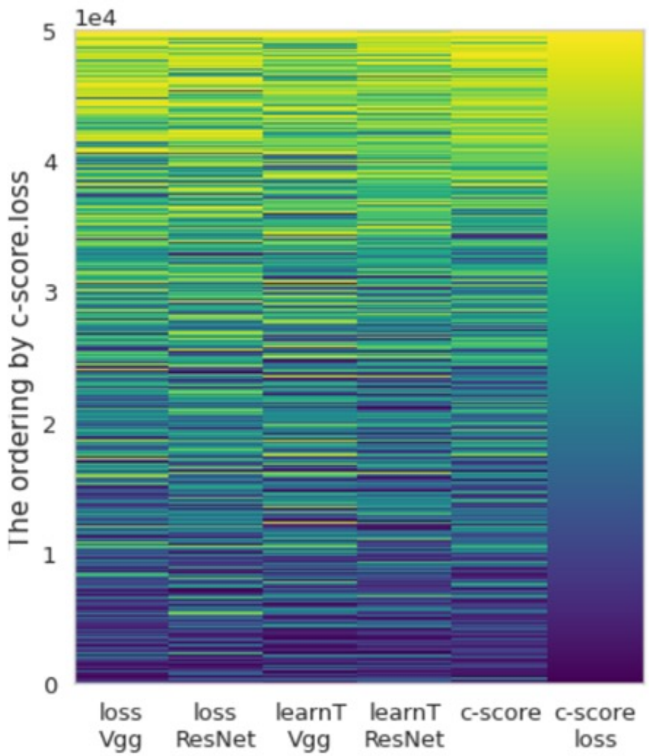
b: the fraction of the training set used at the start of training

Pacing Functions

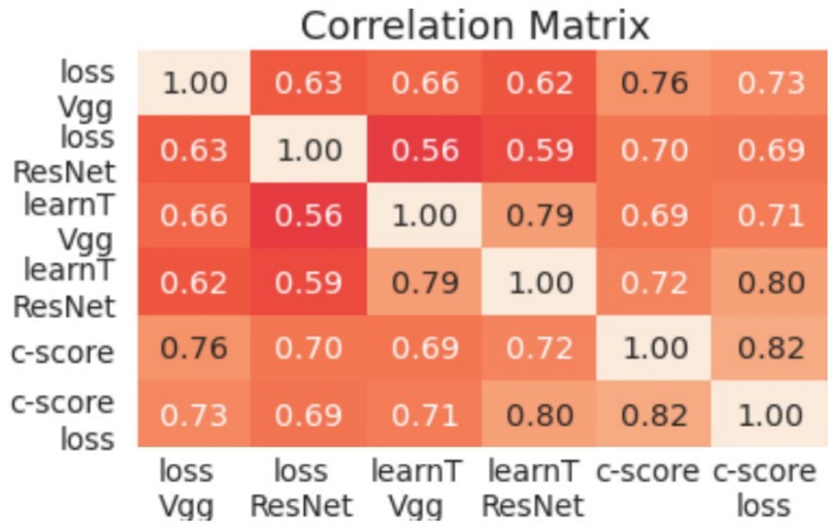
Three Ingredients: Scoring Functions

- Scoring Function
  - Loss function $s(\boldsymbol{x}_i, y_i) = \ell(f_{\mathbf{w}}(\boldsymbol{x}_i), y_i)$.
    - Samples are scored using the real-valued loss of a reference model
  - Learned epoch/iteration $s(\boldsymbol{x}_i, y_i) = \min_{t^*}\{t^* | \hat{y}_{\mathbf{w}}(t)_i = y_i, \forall t^* \leq t \leq T\}$
    - The epoch/iteration for which the model correctly predicts the sample and all subsequent epochs
  - Estimated c-score $s(\boldsymbol{x}_i, y_i) = \mathbb{E}_{D \overset{n}{\sim} \hat{\mathcal{D}} \setminus \{(\boldsymbol{x}_i, y_i)\}}[\mathbb{P}(\hat{y}_{\mathbf{w},i} = y_i | D)]$ where D, with $|D| = n$
    - consistency of a reference model correctly predicting a particular example's label when trained on independent i.i.d. draws of a fixed-sized dataset not containing that example.

Three Ingredients: Scoring Functions



CIFAR10 (epoch 10)

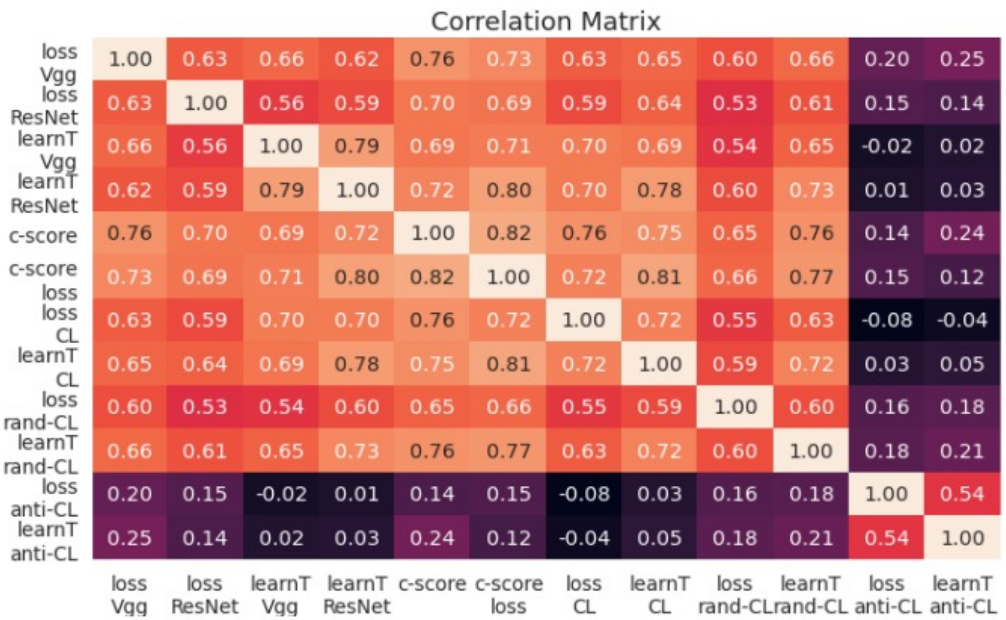


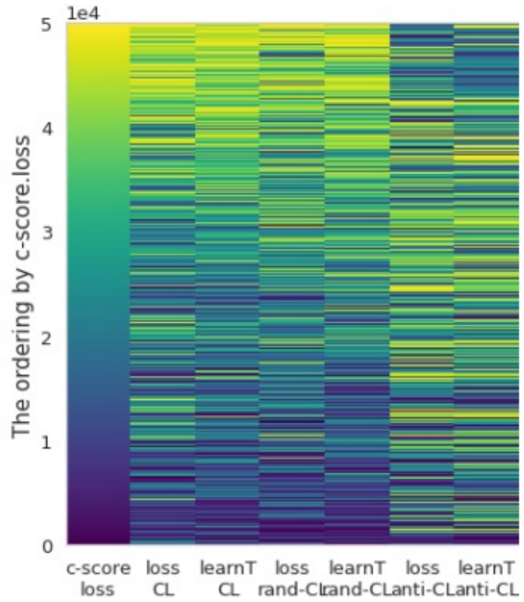
Epoch: {2, 10, 30, 60, 90, 120, 150, 180, 200}
ResNet-18(200 epoch) is only outlier

DIFFICULTY SCORES ARE BROADLY CONSISTENT

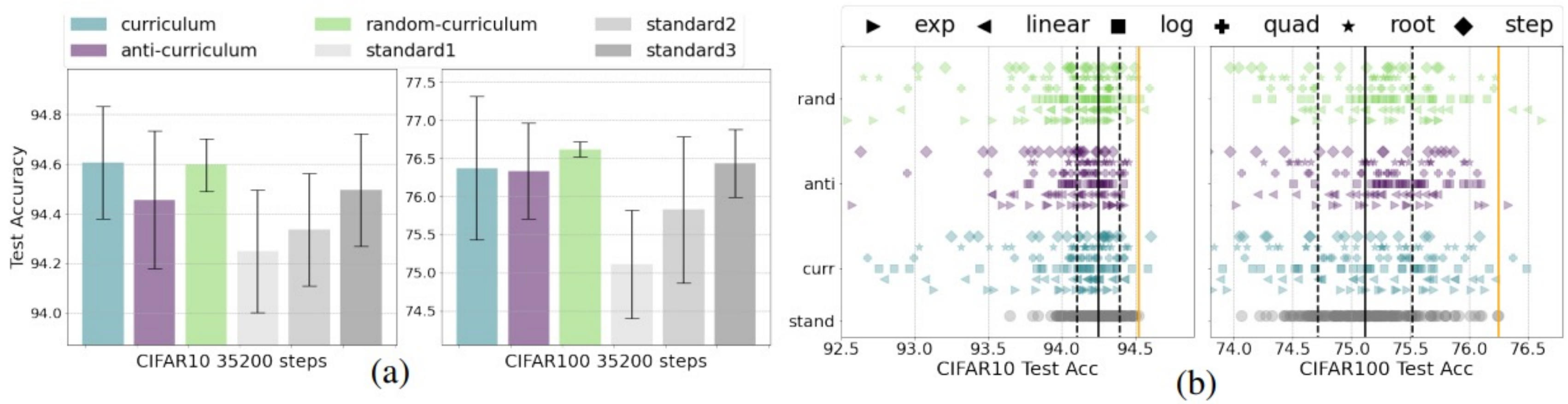
Three ingredients: Pacing Functions

"Does curriculum learning learn the dataset in the order we intended?"



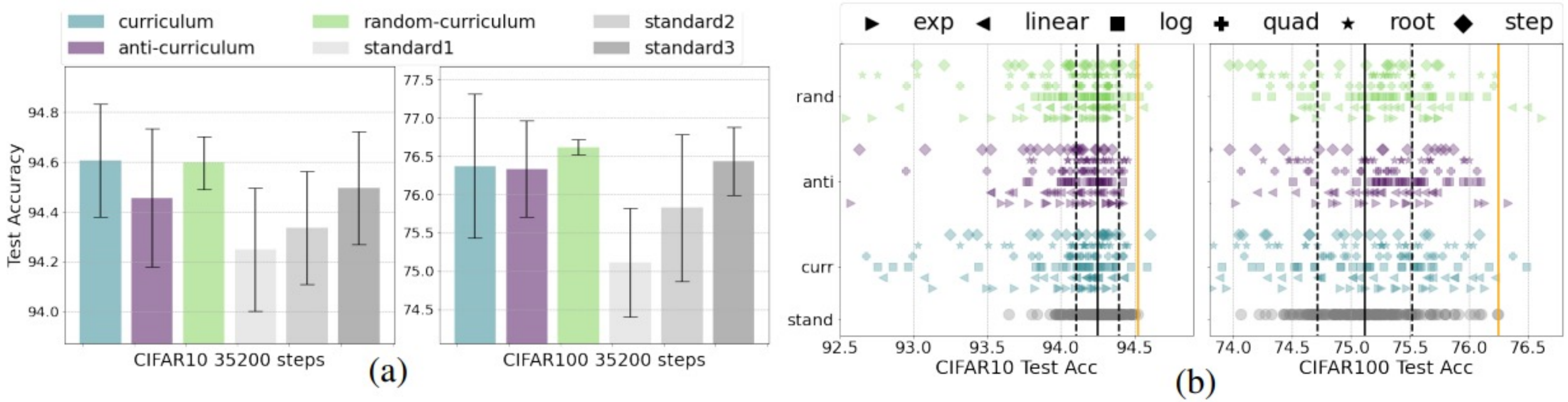


Does Curricula give benefits?



- Standard1: mean performance over all 540 runs
- Standard2: split the 540 runs into 180 groups of 3; the maximum mean
- Standard3: the mean value of the top three values of all 540 runs

Does Curricula give benefits?



- Marginal value of ordered learning
  - An artifact of the large search space
- No dependence on the three different orderings
  - In CIFAR10, the best **mean accuracy** is achieved via **random ordering**
  - In CIFAR100, the best single run has a random ordering

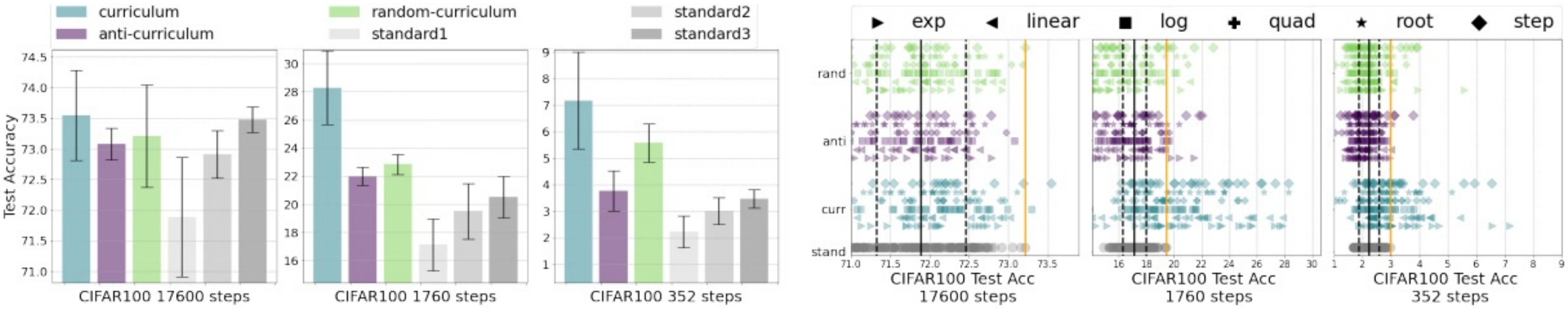
Curricula For Short-Time Training And Noisy Data

- Many large-scale text models are trained using curricula
  - Data-rich setting(multiple epochs of training is not feasible)
  - Data is far less clean than standard image benchmarks.

Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

Table 2.2: Datasets used to train GPT-3. "Weight in training mix" refers to the fraction of examples during training that are drawn from a given dataset, which we intentionally do not make proportional to the size of the dataset. As a result, when we train for 300 billion tokens, some datasets are seen up to 3.4 times during training while other datasets are seen less than once.

Limited training time budget



- Curriculum learning can indeed improve performance when training time budget is decreased

Data With Noisy Labels

- Generating artificial label noise by randomly permuting the labels
  - Recompute the c-score

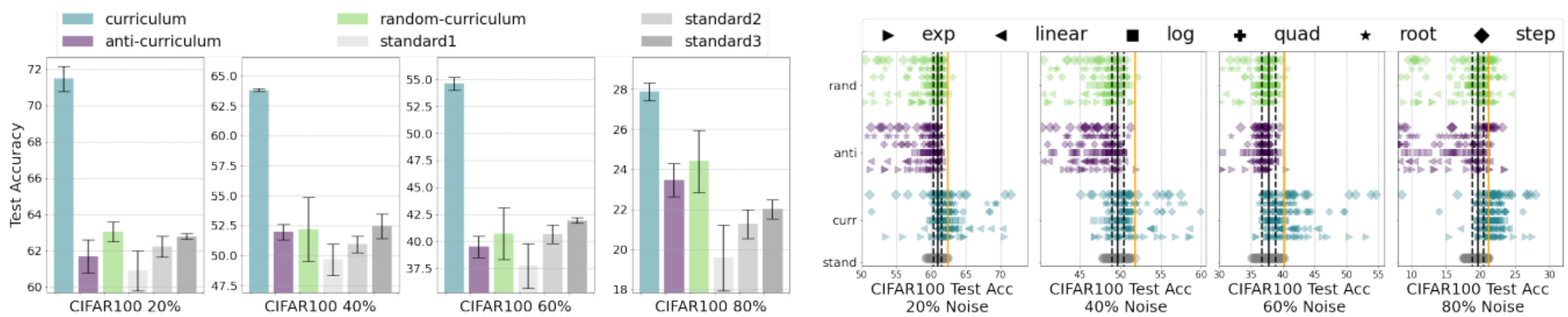
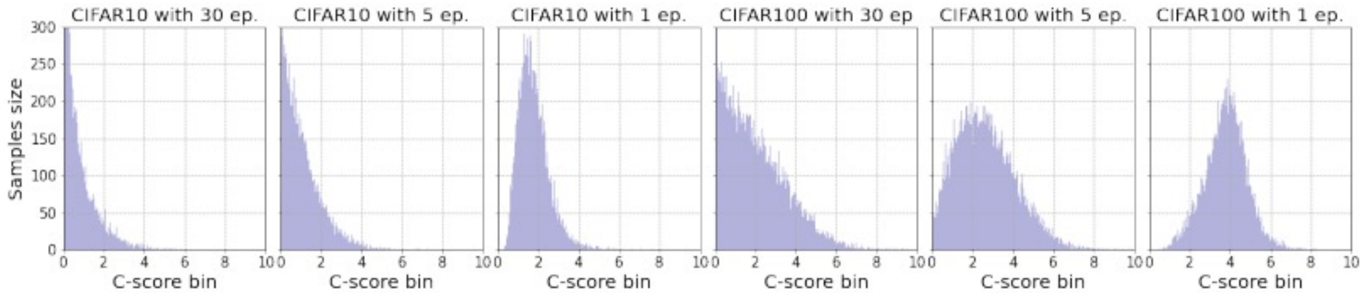
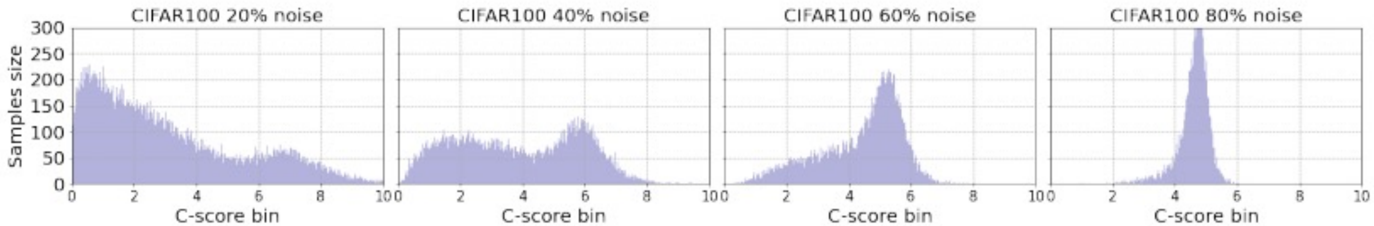


Figure 7: **Curriculum-learning helps when training with noisy labels.** Performance on CIFAR100 with the addition of 20%, 40%, 60% and 80% label noise shows robust benefits when using curricula.

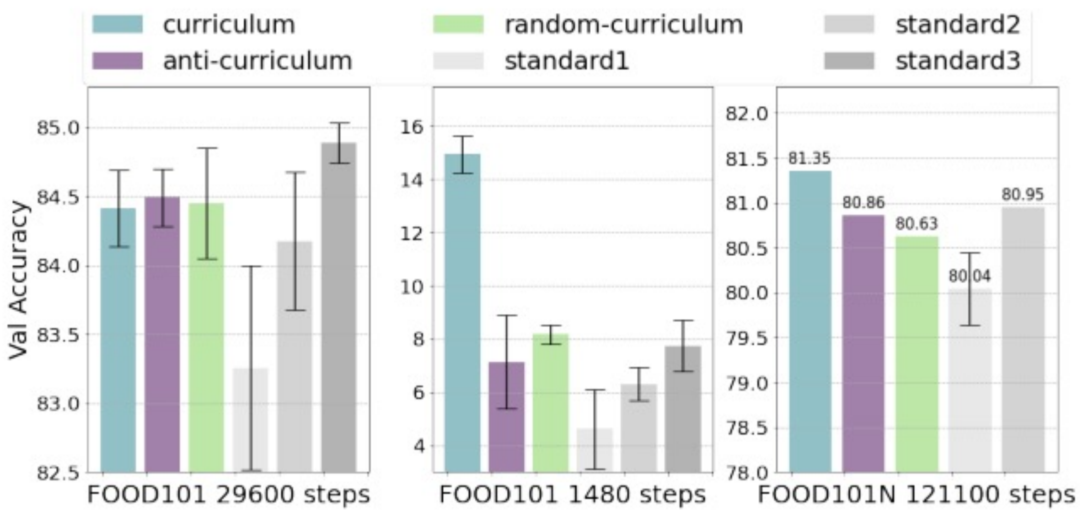
Data With Noisy Labels



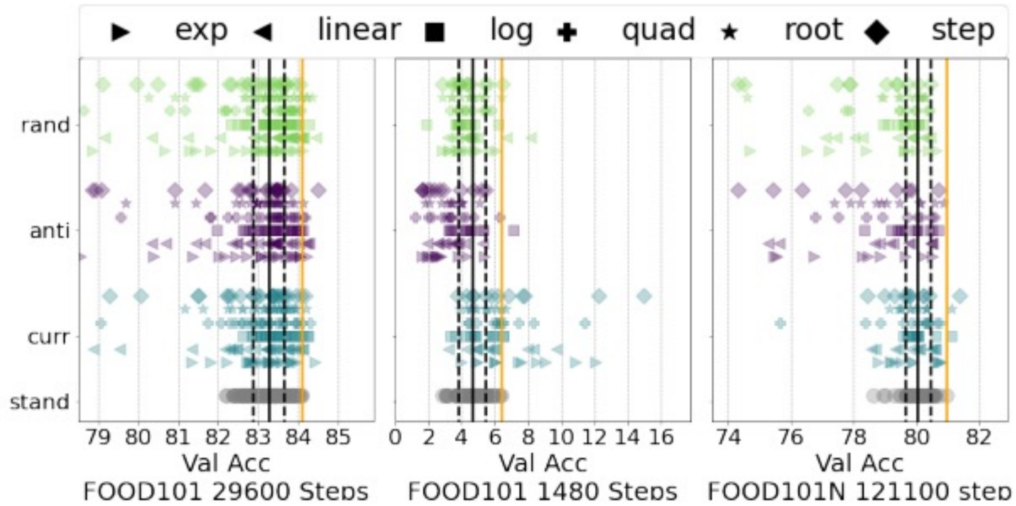


- Why does the label noise benefit from curricula?
    - A significant number of images concentrate around zero (clean CIFAR)

Curricula In The Large Data Regime



(a)



(b)

- FOOD101(75,000 training/25,000 validation)
- FOOD101N(310, 000 training/25,000 validation) - Noisy

Conclusion

- Implicit Curricula: Examples are learned **in a consistent order** (similar architecture, method)
  - And we can change this order by changing the order in which examples are presented during training
- Curricula achieve (almost) **no improvement** in the **standard setting**
- Curriculum learning **improves** over standard training when training **time is limited**
- Curricula improves over standard training in **noisy regime**